

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

Bộ môn: Công nghệ Thông tin.

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC

LẬP TRÌNH PYTHON

Sinh viên: Nguyễn Tuấn Anh.

Mssv:K225480106002.....

Lớp:K58 KTP

Giáo viên GIẢNG DẠY:T.S..Nguyễn Văn Huy.....

Link GitHub: https://github.com/ntanh002-pixel/Bai_Tap_Lon_PYTHON

Thái Nguyên – 2025

BÀI TẬP KẾT THÚC MÔN HỌC
MÔN HỌC: LẬP TRÌNH PYTHON
BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Sinh viên:.....Nguyễn Tuấn Anh

Lớp:..... K58 KTP Ngành:Kỹ Thuật Phần Mềm

Giáo viên hướng dẫn: ...T.S..Nguyễn Văn Huy

Ngày giao đề:20/5/2025..... Ngày hoàn thành:.....9/6/2025.....

Tên đề tài : Trivia Challenge có GUI & file I/O

Yêu cầu :

Đọc file, bắt lỗi file không tồn tại hoặc format sai

GUI: Label câu hỏi, Entry đáp án, nút “Nộp”, Label điểm.....

Tính điểm, chuyển câu hỏi kế tiếp

Nút “Kết thúc” hiển thị tổng điểm

.....

.....

.....

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày....tháng.....năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

MỤC LỤC

CHƯƠNG I: GIỚI THIỆU	7
1: Giới thiệu về đề tài.....	7
2: Tính năng của chương trình	7
3: Thách thức của chương trình	8
4: Vận dụng.....	8
CHƯƠNG II: CƠ SỞ LÝ THUYẾT	9
2.1: Xử lý file văn bản trong Python	9
2.2. Danh sách (List), Tuple và xử lý chuỗi (String)	9
2.3. Hàm (Function) và chia module chương trình.....	10
2.4. Lập trình hướng đối tượng (OOP)	10
2.5. Tkinter – Thư viện giao diện đồ họa	10
2.6. So sánh điều kiện và tính điểm	11
2.7. Vòng lặp và chuyển câu hỏi.....	11
2.8. Cấu trúc khởi chạy chương trình	11
CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH.....	12
3.1. Sơ đồ khối hệ thống.....	12
3.2. Sơ đồ khối các thuật toán chính.....	13
3.3. Cấu trúc dữ liệu.....	16
3.4. Chương trình	16
CHƯƠNG 4 : THỰC NGHIỆM VÀ KẾT LUẬN	18
4.1. Thực nghiệm	18
4.1.1. Đọc dữ liệu từ file:.....	18
4.1.2. Hiện thị giao diện người dùng:	18
4.1.3. Tính điểm và phản hồi kết quả:	18
4.1.4. Kết thúc và thoát:.....	18
4.2. Kết luận	22

4.2.1. Kết quả đạt được:	22
4.2.2. Kiến thức học được:	22
4.2.3. Định hướng cải tiến:	22
Link và QR github	23
Link và QR video youtube	23

LỜI NÓI ĐẦU

Trong thời đại mà giao diện người dùng đóng vai trò quan trọng trong trải nghiệm phần mềm, việc chỉ dừng lại ở những dòng lệnh đơn giản trong Python không còn đủ để đáp ứng nhu cầu thực tiễn. Chính vì thế, bài tập “*Trivia Challenge có GUI & file I/O*” không chỉ là một yêu cầu học phần, mà còn là cơ hội để sinh viên chuyển mình từ tư duy lập trình tuyến tính sang việc xây dựng ứng dụng có tương tác thực sự với người dùng.

Thay vì chỉ hiện câu hỏi và nhập đáp án qua terminal, chương trình lần này được nâng cấp với giao diện trực quan bằng Tkinter, kết hợp với khả năng đọc dữ liệu từ file một cách linh hoạt. Đây là một bước tiến quan trọng – nơi mà kỹ thuật xử lý lỗi, luồng điều khiển, cập nhật giao diện và quản lý trạng thái chương trình cần được kết hợp hài hòa.

Đặc biệt, trong quá trình thực hiện bài tập, tôi đã nhìn thấy rõ một điều: một ứng dụng dù đơn giản đến đâu, khi có giao diện đồ họa và phản hồi tương tác, thì việc tổ chức mã nguồn và xử lý tình huống bất ngờ sẽ trở nên quan trọng hơn rất nhiều. Không còn là những dòng code “chạy được là xong” – bài tập này đòi hỏi một hệ thống thực sự “thân thiện với người dùng”.

Thông qua bài làm này, tôi không chỉ luyện tập kỹ năng đọc/ghi file, xử lý chuỗi hay sử dụng Tkinter, mà còn rèn luyện thêm kỹ năng tổ chức chương trình và tư duy hướng sự kiện – những yếu tố cần thiết cho một lập trình viên phần mềm trong tương lai.

CHƯƠNG I: GIỚI THIỆU

1: Giới thiệu về đề tài

Bài tập lớn "Trivia Challenge có GUI & file I/O" được xây dựng nhằm mục tiêu vận dụng tổng hợp các kiến thức đã học trong môn Lập trình Python, bao gồm: xử lý tệp văn bản, tổ chức dữ liệu, lập trình hướng đối tượng và thiết kế giao diện người dùng với Tkinter.

Chương trình mô phỏng một trò chơi hỏi đáp (Trivia), trong đó người dùng sẽ lần lượt trả lời các câu hỏi được đọc từ một tệp văn bản theo định dạng cho trước. Mỗi câu trả lời sẽ được chấm điểm, và tổng điểm sẽ được hiển thị khi kết thúc trò chơi. Ứng dụng được trang bị các chức năng xử lý lỗi (tệp không tồn tại, định dạng sai), giao diện đơn giản thân thiện, và cơ chế điều khiển luồng câu hỏi – đáp rõ ràng.

Đây là một bài tập có tính thực tiễn cao, không chỉ giúp sinh viên rèn luyện kỹ năng lập trình logic và thao tác với dữ liệu, mà còn tạo nền tảng cho việc phát triển các ứng dụng có giao diện tương tác – một bước quan trọng trong hành trình học lập trình phần mềm chuyên nghiệp.

2: Tính năng của chương trình

Bài tập lớn "Trivia Challenge có GUI & file I/O" yêu cầu sinh viên xây dựng một ứng dụng trò chơi hỏi đáp bằng Python với giao diện đồ họa sử dụng thư viện Tkinter. Ứng dụng sẽ đọc danh sách câu hỏi từ một tệp văn bản định dạng chuẩn (gồm câu hỏi, 3 đáp án và đáp án đúng), cho phép người dùng nhập câu trả lời, tính điểm và hiển thị kết quả sau mỗi lượt chơi. Một số tính năng chính bao gồm: đọc file dữ liệu, xử lý lỗi khi file không tồn tại hoặc định dạng sai, giao diện hiển thị câu hỏi và ô nhập đáp án, tính điểm cho câu trả lời đúng, tự động chuyển sang câu hỏi tiếp theo và cuối cùng là nút "Kết thúc" để hiện tổng điểm.

3: Thách thức của chương trình

Trong quá trình phát triển chương trình, người làm phải đối mặt với nhiều thách thức, tiêu biểu như xử lý đúng cấu trúc file (5 dòng cho mỗi câu hỏi), bắt lỗi định dạng sai, cập nhật điểm một cách động, đảm bảo chương trình chuyển câu chính xác và hiển thị điểm hợp lý. Ngoài ra, việc thiết kế giao diện rõ ràng, dễ sử dụng và tách biệt các thành phần logic (đọc file, GUI, điều khiển) cũng là yêu cầu quan trọng để đảm bảo chương trình chạy ổn định và dễ bảo trì.

4: Vận dụng

Để hoàn thành bài tập, người học cần vận dụng nhiều kiến thức lập trình Python như thao tác với file (hàm `open()`), xử lý ngoại lệ với `try...except`, sử dụng danh sách, chuỗi, tuple để lưu trữ dữ liệu, chia nhỏ xử lý bằng cách viết hàm (`open_file()`, `next_block()`), tổ chức chương trình bằng lập trình hướng đối tượng (class), và sử dụng các thành phần GUI của Tkinter như Label, Entry, Button, StringVar, messagebox. Cuối cùng, chương trình cần được đóng gói trong hàm `main()` để đảm bảo có thể khởi chạy chuẩn mực theo thông lệ lập trình Python.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

Chương trình Trivia Challenge có GUI & file I/O là một ví dụ điển hình về ứng dụng tổng hợp nhiều kiến thức lập trình Python từ cơ bản đến trung cấp. Để xây dựng được chương trình một cách đầy đủ và ổn định, người thực hiện cần nắm vững nhiều khái niệm về xử lý file, cấu trúc dữ liệu, thiết kế giao diện người dùng và tổ chức mã nguồn theo hướng cấu trúc và hướng đối tượng. Dưới đây là các nội dung chuyên môn chính được sử dụng trong chương trình.

2.1: Xử lý file văn bản trong Python

Việc xử lý dữ liệu đầu vào từ file văn bản là một trong những kỹ năng cơ bản nhưng quan trọng trong lập trình Python. Trong bài này, chương trình đọc dữ liệu từ một file .txt theo định dạng chuẩn: mỗi câu hỏi bao gồm 5 dòng – dòng 1 là nội dung câu hỏi, dòng 2–4 là ba phương án trả lời, dòng 5 là số thứ tự của đáp án đúng.

Python cung cấp các hàm và cú pháp tiện lợi để thao tác với file như `open()`, `readline()`, `readlines()`, và cấu trúc `with open(...)` as f: để đảm bảo file được đóng đúng cách. Ngoài ra, cần kết hợp xử lý chuỗi (`strip()`, `split()`) để làm sạch và trích xuất nội dung cần thiết từ từng dòng.

Đặc biệt, chương trình phải có khả năng xử lý lỗi khi file không tồn tại hoặc không đúng định dạng, thông qua khối `try...except`. Việc này không chỉ giúp chương trình ổn định hơn mà còn nâng cao khả năng xử lý ngoại lệ trong thực tế.

2.2. Danh sách (List), Tuple và xử lý chuỗi (String)

Dữ liệu từ file sau khi đọc sẽ được lưu vào các biến dạng danh sách (list) để tiện cho việc truy cập và xử lý. List trong Python là cấu trúc dữ liệu động, cho phép chứa nhiều kiểu dữ liệu khác nhau và có thể thay đổi kích thước linh hoạt. Các đáp án, danh sách câu hỏi, điểm số, và trạng thái trò chơi đều được lưu trữ dưới dạng danh sách.

Tuple – một loại cấu trúc dữ liệu bất biến – cũng có thể được sử dụng để truyền nhóm dữ liệu (ví dụ: một block gồm câu hỏi và đáp án) từ hàm `next_block()` về giao diện.

Chuỗi (string) là kiểu dữ liệu được sử dụng rất nhiều để hiển thị câu hỏi, đáp án và nội dung thông báo. Các phương thức như `.lower()`, `.strip()`, `.format()` hoặc f-string giúp chuẩn hóa và định dạng đầu ra rõ ràng hơn.

2.3. Hàm (Function) và chia module chương trình

Việc chia chương trình thành các hàm nhỏ giúp giảm độ phức tạp và tăng khả năng tái sử dụng mã nguồn. Trong chương trình, các hàm như `open_file()` để mở file, `next_block()` để lấy câu hỏi tiếp theo, `check_answer()` để kiểm tra đáp án... được xây dựng nhằm chia tách chức năng rõ ràng.

Ngoài ra, việc chia module – ví dụ tách phần giao diện và phần xử lý dữ liệu ra các tệp Python riêng – giúp chương trình dễ bảo trì và mở rộng sau này.

2.4. Lập trình hướng đối tượng (OOP)

Python hỗ trợ đầy đủ lập trình hướng đối tượng với các từ khóa `class`, `self`, `__init__`, và các phương thức (method) đi kèm. Trong chương trình này, một lớp chính như `TriviaGUI` được xây dựng để quản lý toàn bộ giao diện và luồng xử lý của trò chơi.

Việc tổ chức chương trình bằng class giúp đóng gói dữ liệu và hành vi thành một khối logic, dễ quản lý, dễ mở rộng và phù hợp với lập trình ứng dụng thực tế. Đây cũng là bước chuẩn bị để người học tiếp cận các framework lớn hơn trong tương lai như PyQt, Django hay Flask.

2.5. Tkinter – Thư viện giao diện đồ họa

Tkinter là thư viện GUI (Graphical User Interface) tiêu chuẩn của Python. Nó cung cấp các widget như `Label`, `Entry`, `Button`, `Frame`, `StringVar`, và `messagebox`, cho phép xây dựng các ứng dụng có giao diện tương tác. Trong chương trình:

`Label` dùng để hiển thị câu hỏi và điểm.

Entry cho phép người chơi nhập câu trả lời.

Button xử lý sự kiện "Nộp" hoặc "Kết thúc".

StringVar() là biến liên kết giữa dữ liệu chương trình và nội dung hiển thị trên GUI.

messagebox.showinfo() dùng để thông báo tổng điểm khi trò chơi kết thúc.

Việc thiết kế GUI yêu cầu tư duy bố trí không gian, luồng xử lý sự kiện (event handling), và đồng bộ dữ liệu giữa giao diện và logic xử lý.

2.6. So sánh điều kiện và tính điểm

Phần quan trọng nhất của trò chơi là kiểm tra đáp án và tính điểm. Điều này được thực hiện bằng cách so sánh chuỗi người dùng nhập vào với đáp án đúng. Việc xử lý không phân biệt hoa thường (sử dụng `.lower()` hoặc `.strip()`) là cần thiết để nâng cao trải nghiệm người dùng.

Điểm được cộng theo quy tắc đơn giản: mỗi câu đúng +1 điểm, sai không cộng. Tổng điểm được hiển thị ở mỗi bước và cuối trò chơi. Cấu trúc `if...else` hoặc `match...case` (trong Python 3.10+) là công cụ chính cho xử lý điều kiện.

2.7. Vòng lặp và chuyển câu hỏi

Sau khi người dùng nộp câu trả lời, chương trình cần chuyển sang câu hỏi tiếp theo. Điều này được thực hiện bằng cách lấy block tiếp theo từ file và cập nhật nội dung các widget tương ứng. Nếu đã đến câu hỏi cuối cùng, nút “Kết thúc” sẽ được bật hoặc chương trình sẽ tự động hiện điểm.

2.8. Cấu trúc khởi chạy chương trình

Python cho phép kiểm soát quá trình khởi chạy chương trình thông qua điều kiện `if __name__ == "__main__":`. Cấu trúc này giúp đảm bảo rằng các dòng mã khởi tạo GUI chỉ chạy khi tệp được thực thi trực tiếp, và không chạy nếu chương trình được import từ nơi khác. Đây là một quy ước chuẩn, giúp chương trình rõ ràng, an toàn và có thể tái sử dụng tốt hơn.

CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. Sơ đồ khối hệ thống

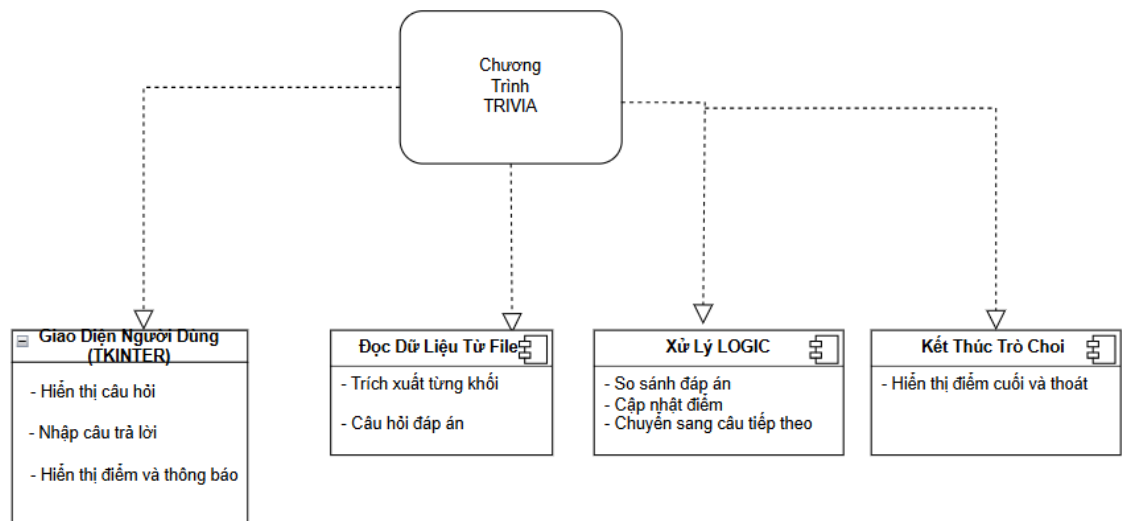
Chương trình Trivia Challenge được thiết kế thành các module riêng biệt, nhằm tách biệt rõ ràng các chức năng và giúp việc phát triển, kiểm thử dễ dàng hơn. Cụ thể, hệ thống bao gồm ba module chính:

Module đọc dữ liệu từ file: thực hiện mở và đọc tệp văn bản chứa các câu hỏi, được định dạng theo chuẩn 5 dòng cho mỗi câu hỏi (gồm nội dung câu hỏi, ba đáp án và đáp án đúng).

Module xử lý logic: kiểm tra câu trả lời của người dùng, tính điểm, quyết định chuyển sang câu tiếp theo hoặc kết thúc trò chơi.

Module giao diện người dùng (GUI): sử dụng thư viện tkinter để hiển thị giao diện, bao gồm các nhãn câu hỏi, ô nhập đáp án, nút điều khiển và hiển thị điểm.

Sự kết hợp giữa ba module trên tạo thành một hệ thống hoàn chỉnh vừa có tính trực quan, vừa đảm bảo tính logic trong xử lý dữ liệu và tương tác với người dùng.



Hình 3.1: Sơ đồ phân cấp chức năng

3.2. Sơ đồ khối các thuật toán chính

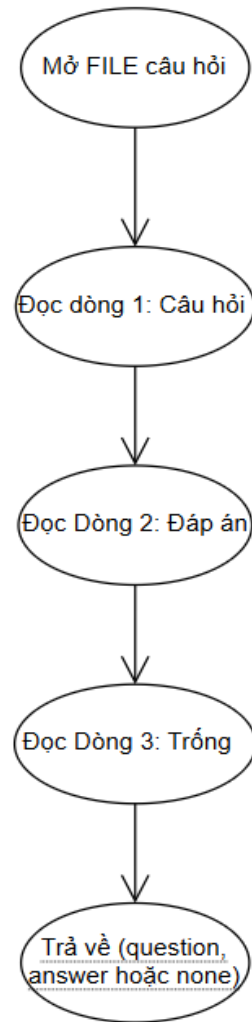
Các thuật toán trong chương trình được tổ chức thành các khối chức năng chính, mỗi khối đảm nhận một vai trò cụ thể trong quá trình chơi:

Thuật toán đọc dữ liệu: thực hiện trong hàm `next_block()`, có nhiệm vụ đọc 5 dòng một lần từ tệp văn bản, trích xuất câu hỏi, các lựa chọn và đáp án đúng. Hàm đảm bảo trả về giá trị hợp lệ và thông báo lỗi nếu định dạng file sai.

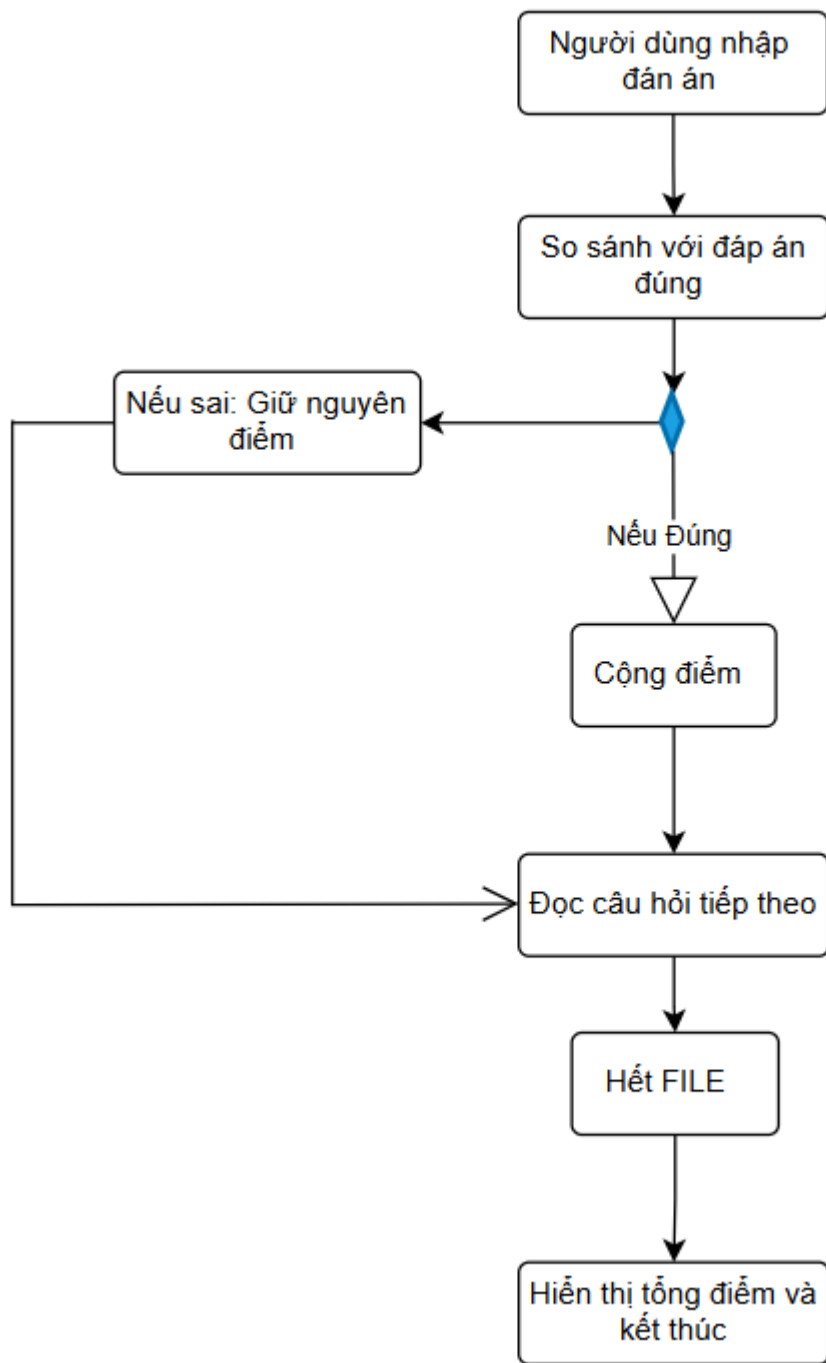
Thuật toán xử lý câu trả lời: so sánh dữ liệu người dùng nhập vào với đáp án đúng. Nếu chính xác, điểm sẽ được tăng thêm 1. Sau đó chương trình tự động chuyển sang câu tiếp theo bằng cách gọi `update_question()`.

Thuật toán kết thúc: được gọi khi không còn câu hỏi nào hoặc khi người dùng nhấn nút "Kết thúc". Thuật toán hiển thị tổng điểm đạt được thông qua `messagebox.showinfo()` và đóng chương trình nếu cần.

Các thuật toán này được viết rõ ràng, chia nhỏ thành các hàm riêng biệt nhằm tối ưu hóa việc tổ chức mã nguồn và tăng khả năng mở rộng sau này.



Hình 3.2: Sơ đồ khối thuật toán xử lý file câu hỏi



Hình 3.3: Sơ đồ thuật toán xử lý logic chơi

3.3. Cấu trúc dữ liệu

Cấu trúc dữ liệu của chương trình đơn giản nhưng hiệu quả. Dữ liệu đầu vào là một tệp văn bản .txt, chứa danh sách các câu hỏi theo định dạng mỗi câu gồm 5 dòng:

Dòng 1: Nội dung câu hỏi

Dòng 2–4: Ba đáp án lựa chọn

Dòng 5: Chỉ số của đáp án đúng (1, 2 hoặc 3)

Trong chương trình, các biến kiểu string và int được dùng để lưu nội dung câu hỏi, các lựa chọn và đáp án đúng. Danh sách câu hỏi được nạp lần lượt thông qua `next_block()` thay vì lưu trữ toàn bộ một lúc, giúp tiết kiệm bộ nhớ và tăng hiệu quả xử lý.

Các biến tạm thời như `score` để lưu điểm, `answer` để lưu đáp án người dùng nhập, và `question_text`, `choices[]` để cập nhật nội dung hiển thị đều được cập nhật động trong quá trình chương trình chạy.

3.4. Chương trình

Chương trình chính được xây dựng dưới dạng lập trình hướng đối tượng, sử dụng một lớp `TriviaGUI` kế thừa từ `tkinter.Frame`. Lớp này chịu trách nhiệm quản lý toàn bộ giao diện và các chức năng liên quan đến người dùng.

Các hàm chính trong chương trình bao gồm:

`open_file()` – mở file văn bản chứa câu hỏi, trả về đối tượng file hoặc báo lỗi nếu không tìm thấy.

`next_block(file)` – đọc 5 dòng liên tiếp từ file, trả về câu hỏi, ba đáp án và đáp án đúng.

`update_question()` – cập nhật nội dung câu hỏi tiếp theo lên giao diện.

`submit_answer()` – xử lý dữ liệu người dùng nhập, so sánh với đáp án đúng, cộng điểm nếu chính xác, sau đó chuyển tiếp hoặc kết thúc.

`end_game()` – hiển thị tổng điểm và thông báo kết thúc trò chơi.

Tất cả các thành phần được gọi và kiểm soát từ bên trong lớp giao diện. Việc tổ chức mã nguồn theo từng hàm riêng biệt giúp chương trình trở nên dễ đọc, dễ sửa và dễ mở rộng hơn nếu muốn nâng cấp sau này (ví dụ như thêm tính năng đếm thời gian, chọn cấp độ, hoặc lưu kết quả).

CHƯƠNG 4 : THỰC NGHIỆM VÀ KẾT LUẬN

4.1. Thực nghiệm

Sau khi hoàn thiện chương trình Trivia Challenge với giao diện đồ họa bằng thư viện tkinter, em đã tiến hành chạy thử và kiểm tra các tính năng chính như sau:

4.1.1. Đọc dữ liệu từ file:

- File văn bản chứa danh sách câu hỏi và đáp án được chương trình xử lý đúng định dạng.

- Các câu hỏi, 3 lựa chọn đáp án và đáp án đúng được tách ra chính xác.

4.1.2. Hiển thị giao diện người dùng:

- Giao diện hiển thị câu hỏi cùng 3 lựa chọn A, B, C dưới dạng các nút chọn.
- Người dùng chỉ cần click chọn đáp án đúng.

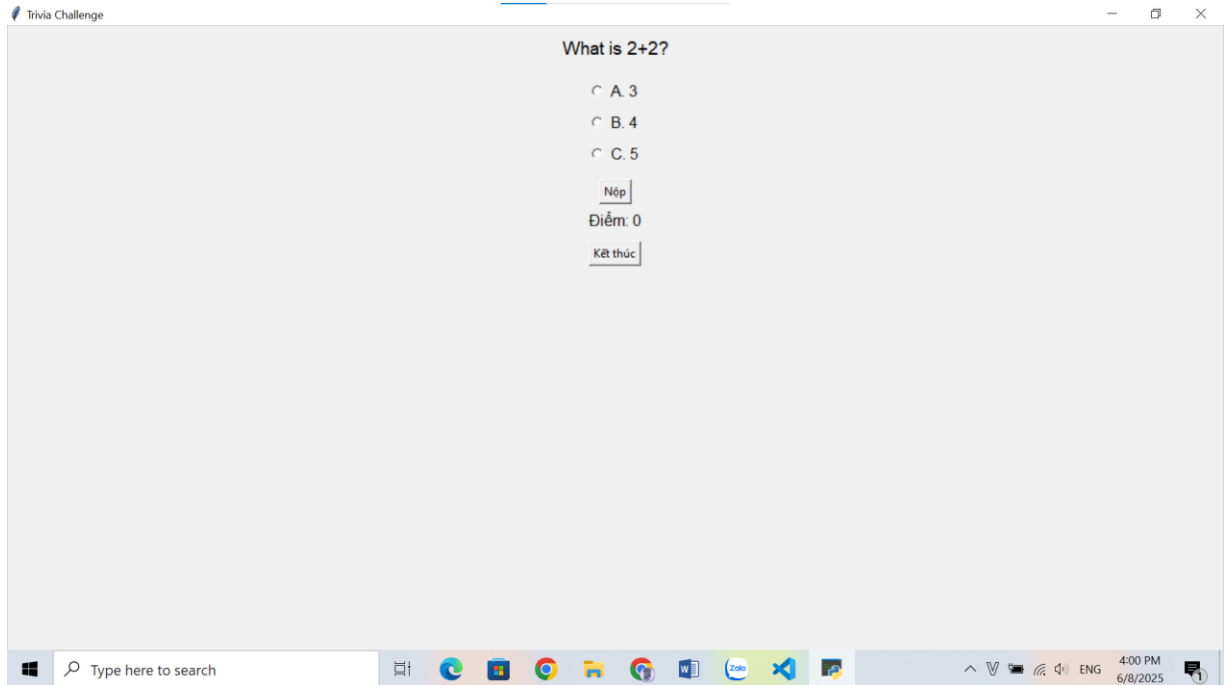
4.1.3. Tính điểm và phản hồi kết quả:

- Sau mỗi lần nộp đáp án, chương trình hiển thị thông báo kết quả (" Đúng" hoặc " Sai") cùng với đáp án đúng nếu chọn sai.

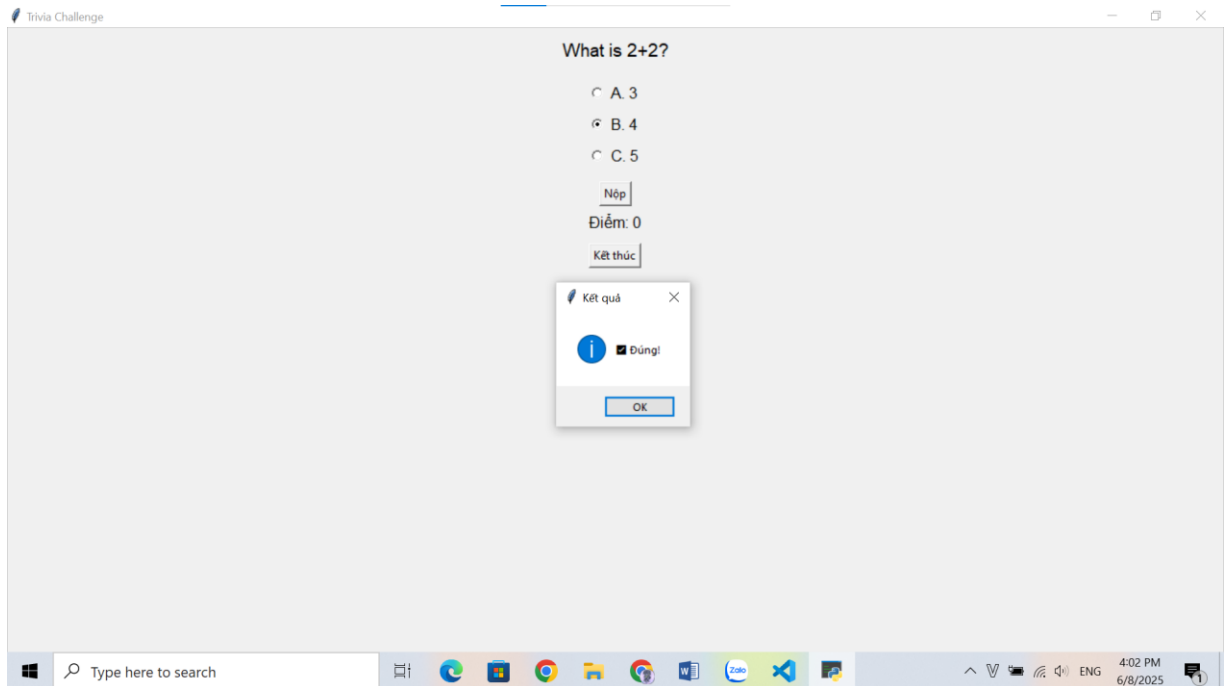
4.1.4. Kết thúc và thoát:

- Khi người dùng hoàn thành tất cả các câu hỏi, chương trình hiển thị điểm tổng kết và tự động đóng cửa sổ khi nhấn nút “Kết thúc”.

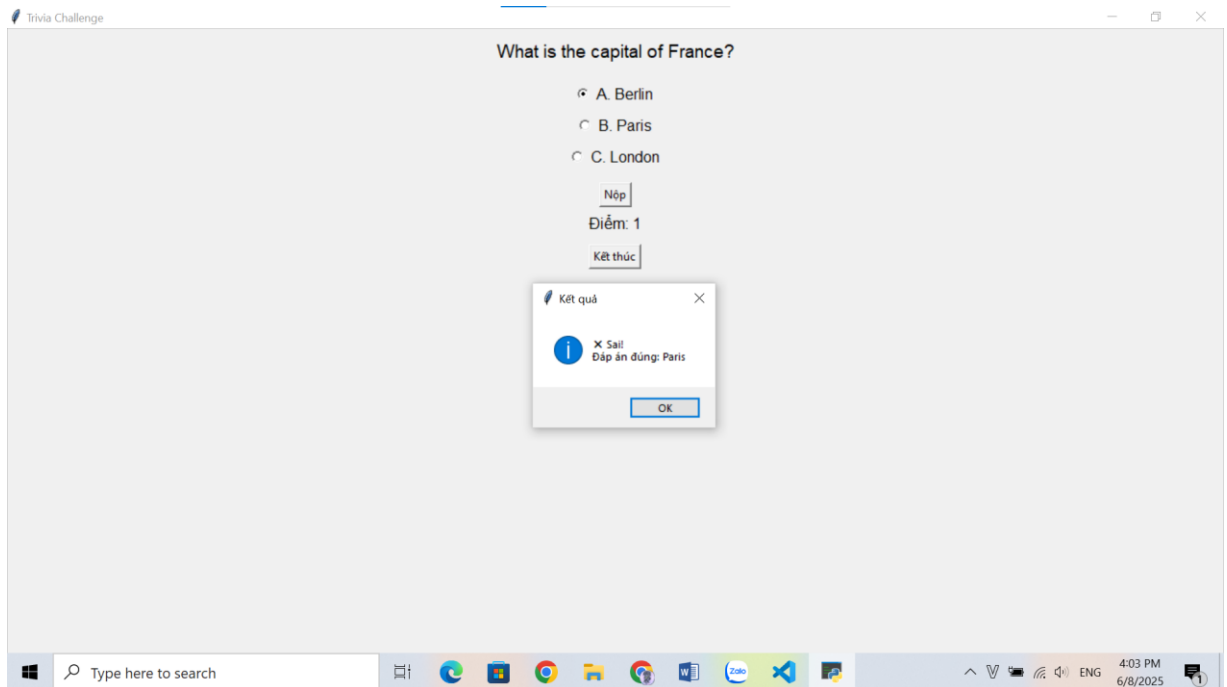
Chụp ảnh lại các màn hình mô tả các tính năng:



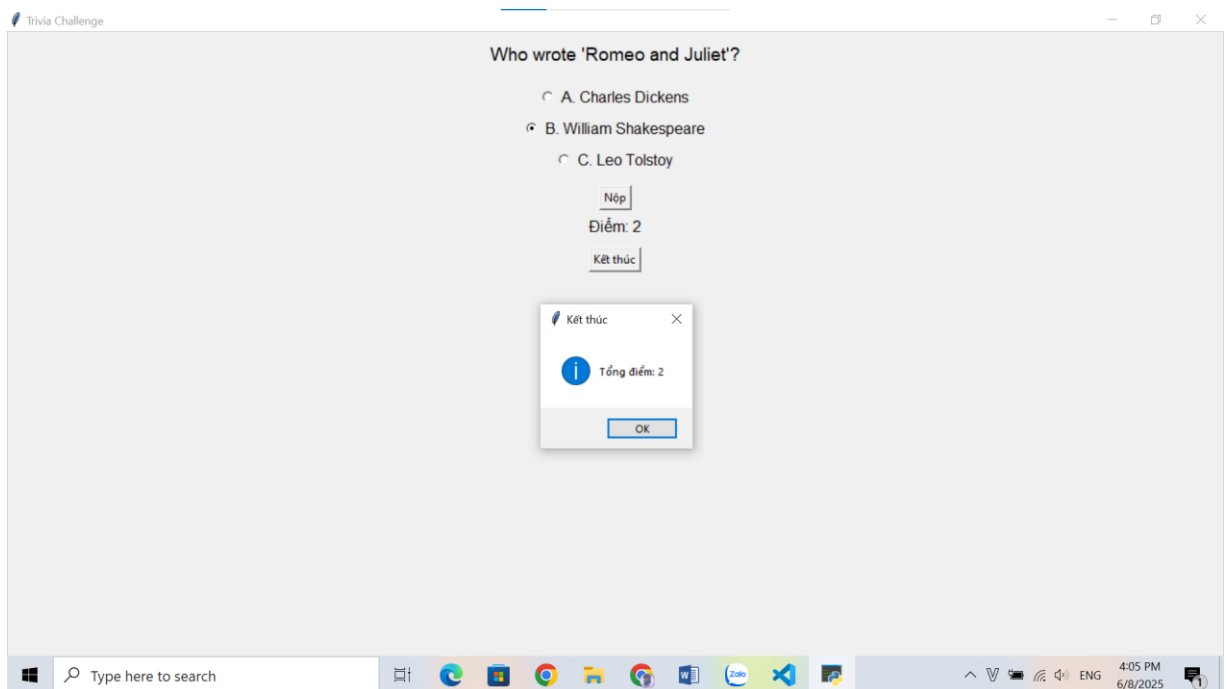
Giao diện lúc bắt đầu câu hỏi



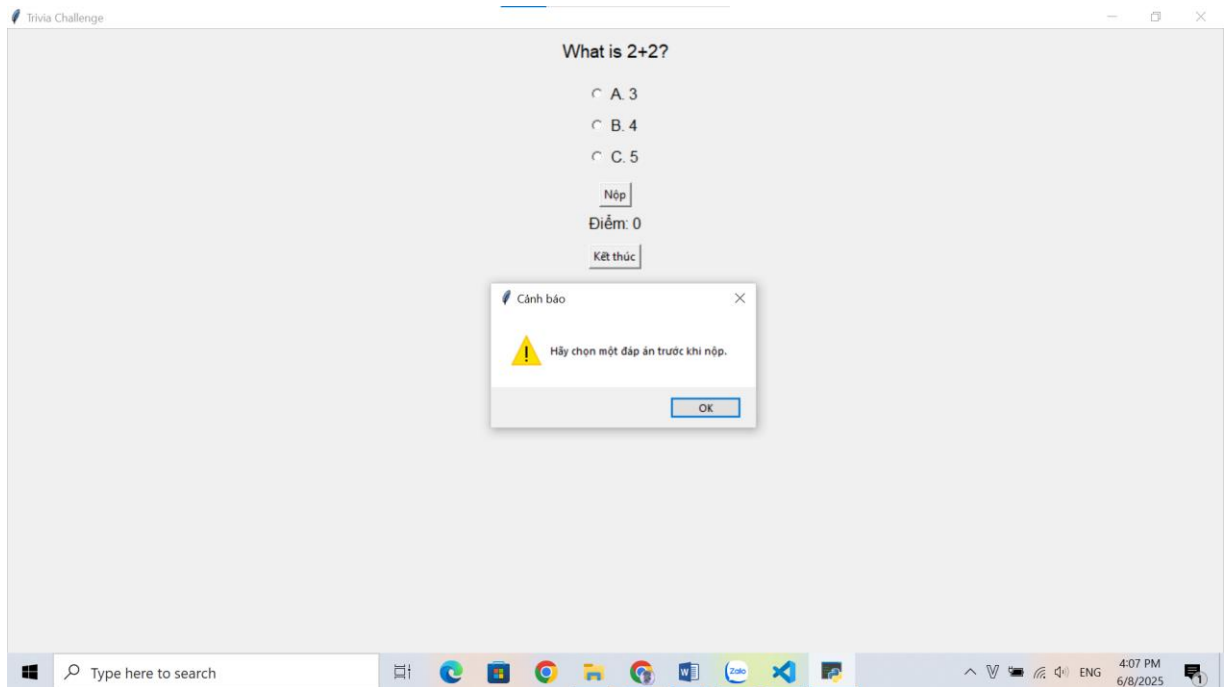
Người dùng chọn đáp án đúng



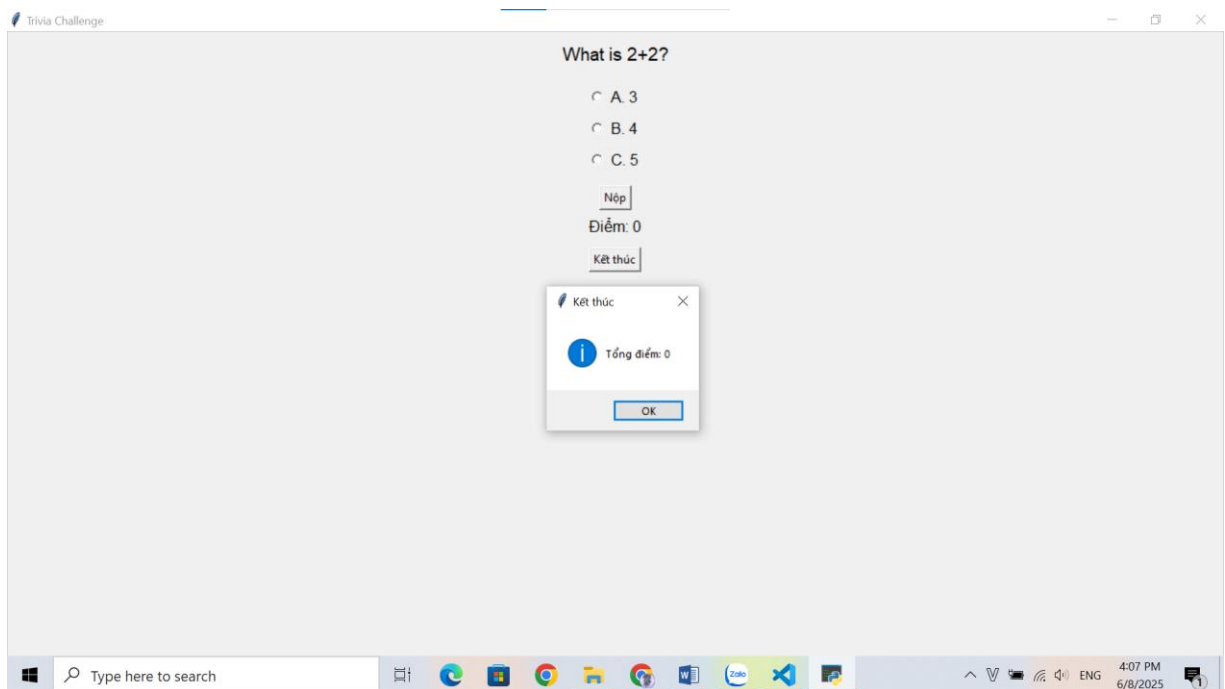
Người dùng chọn đáp án sai và điểm 1 kia là từ đáp án trước



Người dùng chọn đáp án đúng thứ 3 và ấn nộp



Người dùng ấn nộp mà không chọn đáp án



Người dùng ấn kết thúc mà không chọn thì 0 điểm

4.2. Kết luận

4.2.1. Kết quả đạt được:

- Chương trình Trivia Challenge đã hoàn thành với đầy đủ các chức năng chính:
 - + Đọc và xử lý file câu hỏi.
 - + Giao diện người dùng dễ sử dụng và trực quan.
 - + Hiển thị kết quả tương tác rõ ràng, phản hồi nhanh.
 - + Hệ thống tính điểm chính xác và thông báo khi kết thúc.

4.2.2. Kiến thức học được:

- Hiểu rõ hơn về cách:
 - + Tổ chức chương trình Python dạng nhiều module (file_handler.py, trivia_gui.py, main.py).
 - + Sử dụng thư viện tkinter để tạo GUI.
 - + Tạo Radio Button và điều khiển logic qua các hàm sự kiện (command, after...).
 - + Xử lý file văn bản, xử lý lỗi (exception) trong Python.

4.2.3. Định hướng cải tiến:

- Bổ sung thống kê kết quả chi tiết hơn (tỷ lệ đúng/sai, hiển thị lại danh sách câu trả lời).
- Cho phép chọn file câu hỏi khác từ giao diện chính.
- Bổ sung chế độ hẹn giờ cho mỗi câu hỏi.
- Thiết kế lại giao diện với màu sắc sinh động, tương thích nhiều kích thước màn hình hơn.
- Lưu kết quả vào file để theo dõi tiến trình học tập.

Link và QR github

https://github.com/ntanh002-pixel/Bai_Tap_Lon_PYTHON



Link và QR video youtube

<https://youtu.be/Q5HhXJa7Y AQ>



